DeepNews.AI: Detecting Political Bias

Jason Zhao, Abraham Ryzhik, and Nathaniel Lee *
Department of Computer Science
Stanford University
{jzhao23, aryzhik, natelee}@stanford.edu

Abstract

Recent trends in media consumption as well as increasing Internet access to news sources necessitates a reliable and efficient method of detecting biases in the news we consume day-to-day. Without a system in which biases are exposed and controlled, economic incentives will continue to favor news publications and authors that engage in inflammatory and often false journalism. In this paper, we develop and explore two different neural network models that attempt the same classification goal: detecting where each news article lies on the political spectrum from conservative to liberal. The first neural network model takes a convolutional approach and the second is structured with a sequential LSTM (long short-term memory) recurrent neural network (RNN) architecture. For our LSTM RNN's, we design both bidirectional and single directional models. Our results show that CNN's produce the most accurate predictions, and that deeper networks can increase accuracy marginally at the expense of precision and recall.

1 Introduction

With the advent of the Internet Era, where access to information has never been more prevalent, it is essential that we are able to clearly distinguish the biases embedded in the media we consume day to day. We set out to develop an algorithm that can detect the political biases of news articles, rating them on the dimension of political and economic bias (conservative or liberal).

The input to our algorithms is a series of GloVe vectors, with each GloVe vector embedding representing a word in the text of a news article. We then feed this input to both a convolutional neural network model and a sequential LSTM RNN model whose results we will compare. The output for both models is split into two categories: both the ConvNets and the RNN's are trained for binary classification as well as three-class classification. For the binary classification models, a 0 represents an unbiased article, while 1 represents a biased article. The three classes for the ternary classification models are conservative, neutral, and liberal, and the output is a 3D one-hot vector representing these three classes. These classifications represent the output of our models.

2 Related work

The original method of sentiment analysis involves developing sophisticated lexicons designed specifically for political analysis [Young and Soroka, 2012, Ghiassi et al.]. While comprehensive,

* Our political bias detection algorithm is made possible by the support of the DeepNews.AI team led by Stanford's John S. Knight Senior Research Fellow Frederic Filloux and sponsored by the Google European Innovation Fund.

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

these lexicons (which scored words using a point system) also enforced the assumption that each word in a sentence contributes isomorphically to the meaning of a sentence rather that cooperatively with the words surrounding it, leading to frequent misinterpretations of complex phrases with double meanings.

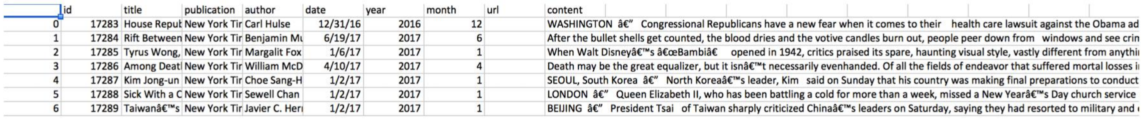
While lexicons offered a simple and comprehensive strategy for conducting basic sentiment analysis, they usually failed to produce human levels of performance as each word holds a plethora of meanings given different contexts, and a single score limited the capability of models to internalize that linguistic variety. Over time, lexicons fell out of favor, to be replaced by a variety of end-to-end machine learning strategies. A group of researchers at Stanford University achieved high training accuracy (nearing 90%) and moderate test accuracy (approximately 60%) using a Naive Bayes Classifier [Moody, 2014].

From recent papers, however, it appears that pure deep learning techniques have emerged as the new cutting-edge of political sentiment analysis. Researchers at Stanford and the University of Maryland independently developed models that achieved accuracies of over 70%, both through the use of RNN's implementing some form of LSTM architecture [Iyyer et al., 2014, Misra and Basak, 2016]. While these results focused on either specific sentences [Iyyer et al., 2014] or congressional debates [Misra and Basak, 2016], they offer an inspiring model for our article analysis system.

Convolutional neural networks (CNN's) are another surprisingly powerful detector for sentiment and bias. Convolutional networks only a few layers deep have produced accuracies of higher than 80% in sentence-level classification tasks with little to no hyperparameter tuning [Kim, 2014]. For this reason, we will also analyze a CNN model in our article bias detector. One factor that both the CNN and RNN approaches have in common is the use of pre-trained word vectors (usually word2vec) that provide a foundation for training. We believe that the vector embedding of textual data is essential for the success and efficiency of our network.

3 Dataset and Features

The dataset used to train our deep learning algorithms is a private dataset collected by Frederic Filloux, our project advisor. It includes tens of thousands of articles from most major news sources: each article is labeled with the title of the articles and the text content of the article.



	id	title	publication	author	date	year	month	url	content
0	17283	House Reput	New York Tir	Carl Hulse	12/31/16	2016	12		WASHINGTON â€” Congressional Republicans have a new fear when it comes to their health care lawsuit against the Obama ad
1	17284	Rift Between	New York Tir	Benjamin Mu	6/19/17	2017	6		After the bullet shells get counted, the blood dries and the votive candles burn out, people peer down from windows and see crin
2	17285	Tyrus Wong,	New York Tir	Margalit Fox	1/6/17	2017	1		When Walt Disneyâ€™s â€œBambiâ€ opened in 1942, critics praised its spare, haunting visual style, vastly different from anythi
3	17286	Among Deatl	New York Tir	William McD	4/10/17	2017	4		Death may be the great equalizer, but it isnâ€™t necessarily evenhanded. Of all the fields of endeavor that suffered mortal losses i
4	17287	Kim Jong-un	New York Tir	Choe Sang-H	1/2/17	2017	1		SEOUL, South Korea â€” North Koreaâ€™s leader, Kim said on Sunday that his country was making final preparations to conduct
5	17288	Sick With a C	New York Tir	Sewell Chan	1/2/17	2017	1		LONDON â€” Queen Elizabeth II, who has been battling a cold for more than a week, missed a New Yearâ€™s Day church service
6	17289	Taiwanâ€™s	New York Tir	Javier C. Herı	1/2/17	2017	1		BEIJING â€” President Tsai of Taiwan sharply criticized Chinaâ€™s leaders on Saturday, saying they had resorted to military and

We processed the text content of each article, first by converting each article into a sequence of numerical tokens representing words. From this numerical vocabulary, we then converted each number (and thus each word) into a 100D GloVe vector through a Keras embedding layer designed to perform this conversion inserted at the beginning of each of our models.

For binary classification models, our train and validation data sets included articles from New York Times, Atlantic, Fox News, and Breitbart labeled as biased and articles from Reuters and CNN labeled as unbiased. For our test set, we expanded the diversity of publications, including articles from New York Post, National Review, NPR, Guardian, and Vox as biased and articles from the Washington Post as unbiased (as well as additional articles from the original publications in the train and validation sets). For three-class classification models, we used the same articles from the same publications, but with a new distinction between conservative, neutral, and liberal. For our train and validation sets, articles from New York Times and Atlantic were labeled as liberal, articles from Breitbart and Fox News were labeled as conservative, and articles from Reuters and CNN were labeled as neutral or unbiased. For our test set, (in addition to the original publications from the train and validation sets), we labeled articles from Vox, NPR, and Guardian as liberal; articles from New York Post and National Review as conservative; and articles from Washington Post as neutral. While ideally every article could be labeled by hand, upon discussing with Frederic (our advisor and a longtime journalist) we determined that the vast majority of articles follow the bias of their publication; furthermore, no op-eds were included in our dataset. Additionally, the expansion of many new publications in our test dataset ensured that we were not simply learning the format of specific publications in the train/dev sets. For either classification, our training set included 52252 articles, our validation set included 13063 articles, and our test set included 19106 articles. In each distribution, similar amounts of articles from each publication were used. Our train/dev/test distribution approximates a 62%/15%/23% split.

4 Methods

All of our learning algorithms were developed using a combination of the Keras and Tensorflow libraries [Abadi et al., 2016, Chollet et al., 2015]. We trained two different types of deep learning neural networks, which we will describe separately. The first is our convolutional neural networks (CNN's). We decided to train CNN's because its use of convolution in place of general matrix multiplication in its convolutional layers allows for local spatial coherence and thus more shared parameters. We trained four convolutional neural networks: one, two, three, and four layer convolutional networks. Each of these four CNN's was trained to perform both binary and three-class classification. Our first layer (after the initial embedding layer used to obtain GloVe vectors) in each CNN was a 1D convolution, followed by a 1D max-pooling layer. This convolution-pooling sequence was repeated for each layer in the CNN's with multiple layers. At the end of each CNN, there was a fully connected (or dense) layer with a ReLu activation ($f(x) = max(0, x)$), finally followed by a Sigmoid activation ($S(x) = \frac{1}{1+e^{-x}}$ in the case of binary classification) or Softmax activation ($p_j = \frac{e^{o_j}}{\sum_k e^{o_k}}$ in the case of three-class classification).

The second is our LSTM recurrent neural networks. We decided to train LSTM RNN's because of their looping architecture, which generates an artificial form of memory that carries information from past layers into future ones, which is useful in text processing. Furthermore, the LSTM feature allows for this information to be passed on for many layers with less risk of memory degradation as each layer runs. We trained two sets of three RNN's. Each set contains one, two, and three layer recurrent networks; the first set contains bidirectional layers (which allow for the model to have both backward and forward information at every step) while the second set contains single directional layers. Each of these RNN's was trained to perform both binary and three-class classification. For each sequential layer in the RNN, we implemented dropout regularization. Each RNN ended with one fully connected layer, and then a final activation function: either Sigmoid activation ($S(x) = \frac{1}{1+e^{-x}}$) or Softmax activation ($p_j = \frac{e^{o_j}}{\sum_k e^{o_k}}$).

We utilized separate loss functions: with binary classification models, we utilized a binary cross-entropy loss function, which can be formalized as: $H_{y'}(y) := -\sum_i (y'_i \log(y_i) + (1 - y'_i)\log(1 - y_i))$. For three-class classification models, we utilized a categorical cross-entropy loss function, which can be formalized as: $H(p, q) = -\sum_{i=1}^{n} p(x_i)\log(q(x_i))$ For all models, we utilized the ADAM optimizer.

5 Experiments/Results/Discussion

We set our learning rate to 0.001 for every model that we tested. This is based off of a standard we saw from related models, and we opted for a common learning rate that would function well across all models as a baseline. We implemented a mini-batch size of 128. We opted for this relatively smaller size because we believed that many articles were likely to be ambiguous in their bias, and thus it was to our advantage to incorporate many finer updates through a mini-batch size that could closely approximate stochastic gradient descent even at the expense of efficiency benefits of vectorization. As discussed in the previous section, we implemented cross-entropy loss functions as opposed to means squared error (MSE) loss because our problem involves classification into discrete categories as opposed to regression over a continuous numerical scale. Due to the logarithmic properties of cross-entropy loss, the gradient does not shrink as fast as we approach values close to 0 or 1. This makes it ideal for classification as we wish to drive values towards extremes as outputs are clearly defined and discrete.

Our primary metric was accuracy: we wanted to ensure that given a large quantity of articles, our political bias detector models could classify a very large proportion of them correctly, as misclassification could lead to potentially jarring and false interpretations by individuals of the content that they are consuming. We also measured precision as well as recall: both are important to our overall measurement of our models: false positives and false negatives both represent essentially the same thing: a misclassification of bias or neutrality. Thus, we will also utilize F1 scores to account for both precision and recall.

Layers:	1	2	3	4
Conv	81.98%	80.66%	81.28%	83.04%
LSTM	81.31%	81.81%	81.79%	N/A
BiDir	81.68%	81.84%	82.07%	N/A

Table 1: Accuracies of 3-Class Models

Layers:	1	2	3	4
Conv	96.56%	96.01%	96.59%	96.60%
LSTM	96.20%	96.41%	96.47%	N/A
BiDir	96.39%	96.44%	96.33%	N/A

Table 2: Accuracies of Binary Models

Analyzing the accuracies from Table 1, we see that for the three-class classification task, the four-layer CNN had the highest accuracy of 83.04%. We believe that the success of the four-layer CNN in particular was a combination of the local "spatial" coherence properties of convolutions as well as the deeper network (no other binary classification model had four layers). However, we see that generally, additional layers for both the bidirectional and single directional RNN's did not increase accuracy by a significant amount, although it did in general lead to some increase in accuracy. We will demonstrate later why the additional layers, despite their marginal positive impacts on accuracy, come with a price.

Analyzing the accuracies from Table 2, we see that for the binary classification task, the four-layer CNN again had the highest accuracy, this time of 96.40%. We believe that the same factors lead the deepest CNN to have the highest accuracy, and we attribute the much higher average accuracy in the binary classification task to the fact that the binary classification is much less nuanced than the three-class classification, which must distinguish between neutral and two different extremes of bias. Furthermore, we observe that additional layers does not increase the accuracy by a notable amount, particularly in the case of the LSTM models. We will demonstrate why we believe that for LSTM models (both bidirectional and single directional), fewer layers is often times a better choice (given our dataset).

Layers:	1	2	3	4
Conv	0.95	0.94	0.94	0.96
LSTM	0.94	0.95	0.96	N/A
BiDir	0.96	0.95	0.95	N/A

Table 3: F1 Scores for Three-Class

Layers:	1	2	3	4
Conv	0.85	0.83	0.82	0.81
LSTM	0.81	0.80	0.82	N/A
BiDir	0.82	0.83	0.81	N/A

Table 4: F1 Scores for Binary

From Table 3, we notice that sometimes F1 decreases as we increase the depth of some of our three-class models. This trend is repeated amongst many classifiers for binary classification (Table 4). We believe that this is an example of overfitting to our dataset, which includes many more examples of biased articles than unbiased articles, making it easier (especially as our model fits to our data better) to simply guess "biased" more often, increasing our accuracy but decreasing our F1 score, and thus decreasing our hypothetical accuracy on a true real world spread. Recognizing this problem, we implemented dropout regularization to mitigate overfitting. It is interesting to note that neutral articles generated the highest average F1 scores, while liberal articles generated the least, indicating that perhaps liberal articles are less distinctive.

Analyzing Figure 1 and Figure 2, we observe that while the binary CNN and single directional LSTM RNN both achieve reasonable ROC's, again, the addition of deeper layers tends to decrease ROC, again pointing to the observation made previously regarding the negative effect of overfitting on precision and recall.

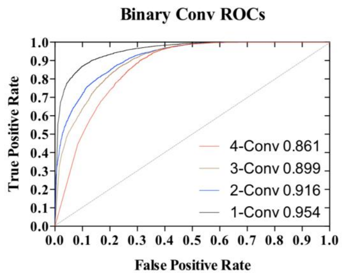


Figure 1

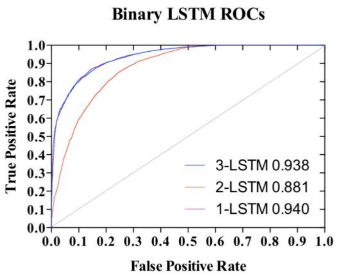


Figure 2

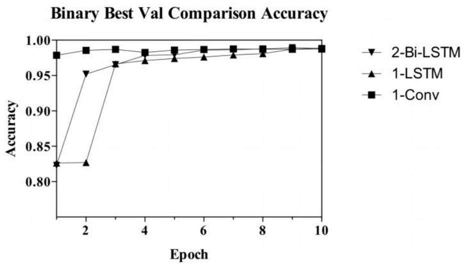


Figure 3

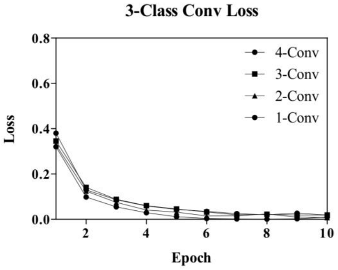


Figure 4

Finally, from Figure 3, we note that in addition to achieving the highest comparison accuracy (amongst the other models in the graph), the binary one-layer convolution also achieves an incredibly high accuracy with little train time. We believe that the LSTM RNN's took longer to train and achieved slightly lower performance due to the length of some articles, which made it difficult for features to be propagated across such lengths. Additionally, from Figure 4, we note that convolutional networks also achieved a steep reduction in loss with few epochs, a feature not common amongst other models we trained. Thus, we believe that overall, a one-dimensional CNN maximizes accuracy benefits without incurring the ROC reductions from additional layering.

As a final analysis, we edited the text of some articles and analyzed subsequent effects of model classification. Removing "CNN" from CNN articles caused models to have trouble predicting CNN as either neutral or liberal, while removing "FOX" from Fox News articles did not have considerable effects on their classification. Interestingly, removing "Bernie Sanders" from articles tended to cause the articles to be rated as less conservative, suggesting that conservative articles spent more time (presumably) criticizing the politician than liberal articles did praising him. It was encouraging to see that our article recognized salient political figures as having an effect on the bias of an article. Finally, reducing the length of an article to a single paragraph had very little effect on classification, suggesting that there are major differences between our classification categories that can be identified without an entire article's worth of text. Reducing an article to a single sentence tended to result in a neutral label, but adding as few as three sentences together was enough for the model to start seeing conservative or liberal bias with high confidence.

6 Conclusion/Future Work

In summary, we discovered that four-layer CNN's were the most accurate bias detectors for our dataset, although LSTM RNN's (both bidirectional and single directional) were quite effective as well. Interestingly, we observed that additional layers could increase the accuracy of model predictions, but often simultaneously lowered precision and recall, perhaps due to imbalances in our dataset. In the future, we hope to explore nuances beyond simple classification of conservative and liberal, instead performing regression to determine spectrums of economic left/right and political authoritarianism/libertarianism, plotting each article on a continuous 2D Cartesian plane. We would also like to combine convolutional layers with LSTM layers, using convolutions before sequential layers to detect features and decrease the time-steps the LSTM's must process.

7 Contributions

Jason Zhao: Worked on formatting and composing the milestone proposal and final project report as well as programs for building the data and the data pipeline. Helped design overall data labeling process and train/test/dev set distributions.
Abraham Ryzhik: Worked on developing the Keras model architecture as well as the data pipeline. Assisted with pre-processing the dataset. Focused on implementation of the CNN model and loading the project on a GPU.
Nathaniel Lee: Worked on developing the TensorFlow model architecture as well as the data pipeline. Assisted with pre-processing the dataset. Focused on the implementation of the RNN model as well as the TensorBoard visualizations.

References

Lori Young and Stuart Soroka. Affective news: The automated coding of sentiment in political texts. *Political Communication*, 29(2):205–231, 2012.

M. Ghiassi, J. Skinner, and D. Zimbra. Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network. *Expert Systems with Applications*.

Rush Moody. Bias detector: Using language models to identify editorial political slant. *Stanford CS229*, page 6, 2014.

Mohit Iyyer, Peter Enns, Jordan Boyd-Graber, and Philip Resnik. Political ideology detection using recursive neural networks. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1113–1122, 2014.

Arkajyoti Misra and Sanjib Basak. Political bias analysis. *CS224n*, page 8, 2016.

Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

François Chollet et al. Keras. `https://keras.io`, 2015.